

Offline messenger

Perju Rares-Andrei

Universitatea Alexandru Ioan Cuza , Facultatea de Informatica Iasi

perju.rares.andrei2001@gmail.com

rares.perju@student.uaic.ro

Abstract: Offline messenger , o aplicatie realizata in limbajul de programare C la nivel client/server ce permite transmiterea de mesaje de la un user la altul.

Keywords: TCP/IP , offline , client , server ,protocol de comunicare ,niveluri ,port,diagrama, baze de date

1. Introducere

Offline messenger este o aplicatie de tip client/server concurenta ce permite conectarea mai multor utilizatori simultan . Acesta permite trimiterea de mesaje catre alti utilizatori inregistrati , indiferent daca receptorul mesajului este sau nu conectat in acel moment la server .

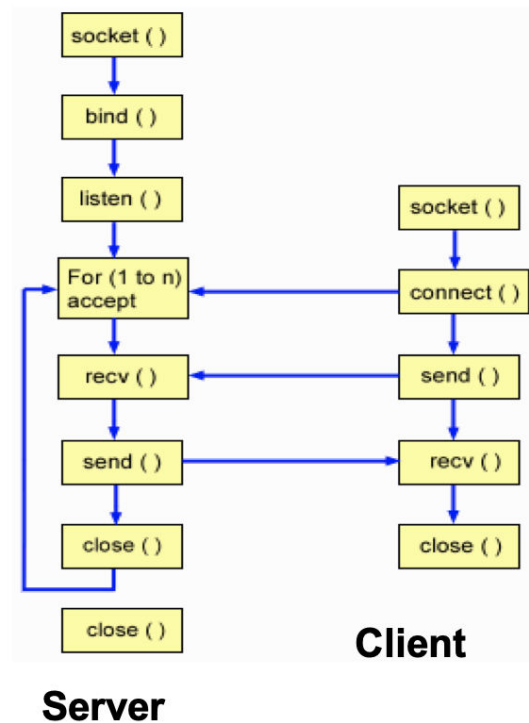
Aceasta aplicatie permite vizualizarea istoricului conversatiilor si posibilitatea de replay al unui mesaj selectat . In cazul in care un utilizator nu este conectat la server si primeste un mesaj acesta are posibilitatea de vizualiza mesajul primit .

Pentru a trimite un mesaj utilizatorul trebuie sa fie logat , trimiterea mesajului fiind realizata de catre server.

2. Tehnologii utilizate

2.1.1. TCP/IP

Tipul de protocol necesar realizarii comunicarii este TCP (Transmission Control Protocol) concurent , deoarece realizeaza o comunicare sigura(datele sunt trimise in pachete acestea fiind transmise independent si sunt reasamblate odata ce ajung la destinatie pentru a ajunge la forma originala) , ordonata(toate pachetele sunt reasamblate in ordine) si fara pierdere de date . Acest protocol permite accesarea simultana de mai multe dispozitive conectate la aceeași rețea .



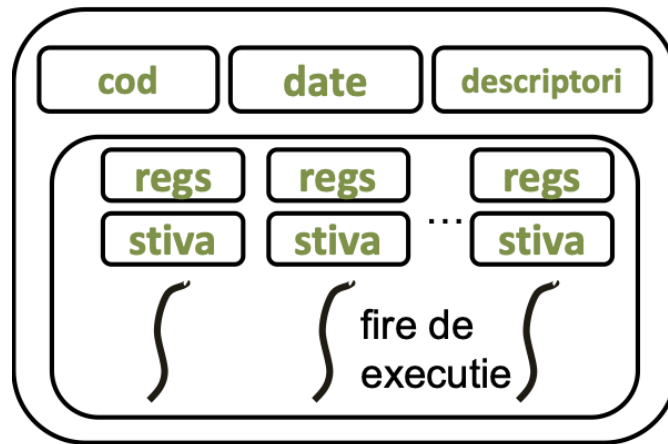
Model server/client TCP

2.1.2. TCP Concurrent

În modelul concurrent, porțiunile de ascultare și server ale aplicației rulează sub controlul diferitelor sarcini. Scopul Ascultătorilor este de a accepta conexiunea și de a invoca sarcina Server. Partea Server a aplicației se ocupă cu trimiterea și primirea datelor aplicației și efectuarea procesării dependente de aplicație. Acest model permite un grad mai mare de concurență a tranzacțiilor, deoarece soclul de ascultare nu este deținut de un singur client și poate asculta simultan mai mulți clienți.

În cele patru modele TCP/IP acceptate de TI, există atât un aspect de ascultare a conexiunii TCP/IP, cât și un aspect de server. Modul în care sunt implementate porțiunile de ascultare și server ale aplicației determină dacă este utilizat modelul de acces iterativ sau concurrent. Modelul de acces simultan necesită utilizarea unui mesaj de cerere de tranzacție (TRM); modelul iterativ nu. TRM este o înregistrare de date formatată care identifică programul de tranzacție (TP) IMS sau CICS care trebuie invocat și caracteristicile acestuia.

De asemenea serverul se bazează pe implementarea sockets multithreading ce oferă conectarea simultană și concurența a mai multor utilizatori.



Procese cu mai multe fire de executie

2.1.3. Niveluri

Diversele cerințe au condus la alegerea a patru niveluri pentru modelul TCP/IP: Aplicație, Transport, Rețea (sau Internet) și Acces la Rețea .



I. Nivelul Aplicație

Nivelul aplicație se referă la protocoalele de nivel înalt folosite de majoritatea aplicațiilor, precum terminalul virtual (TELNET), transfer de fișiere (FTP) și poștă electronică (SMTP). Alte protocoale de nivel aplicație sunt DNS (sistem de nume de domeniu), NNTP sau HTTP.

II. Nivelul Transport

Este identic cu cel din modelul OSI, ocupându-se cu probleme legate de siguranță, control al fluxului și corecție de erori. El este proiectat astfel încât să permită comunicarea între entitățile pereche: sursă, respectiv, destinație. În acest sens au fost definite două protocoale capăt-la-capăt.

TCP (Transmission Control Protocol). El este un protocol sigur orientat pe conexiune care permite ca un flux de octeți trimiși de pe o mașină să ajungă fără erori pe orice altă mașină din inter-rețea. Acest protocol fragmentează fluxul de octeți în mesaje discrete și pasează fiecare mesaj nivelului internet. TCP tratează totodată controlul fluxului pentru a se asigura că un emițător rapid nu inundă un receptor lent cu mai multe mesaje decât poate acesta să prelucreze.

III. Nivelul Rețea (Internet)

Scopul inițial al nivelului rețea ("Internet Protocol") era să asigure rutarea pachetelor în interiorul unei singure rețele. Odată cu apariția interconexiunii între rețele, acestui nivel i-au fost adăugate funcționalități de comunicare între o rețea sursă și o rețea destinație.

În stiva TCP/IP, protocolul IP asigură rutarea pachetelor de la o adresă sursă la o adresă destinație, folosind și unele protocoale adiționale, precum ICMP sau IGMP. Determinarea drumului optim între cele două rețele se face la acest nivel.

Comunicarea la nivelul IP este nesigură, sarcina de corecție a erorilor fiind plasată la nivelurile superioare (de exemplu prin protocolul TCP). În IPv4 (nu și IPv6), integritatea pachetelor este asigurată de sume de control.

IV. Nivelul Acces la rețea

Se ocupă cu toate problemele legate de transmiterea efectivă a unui pachet IP pe o legătură fizică, incluzând și aspectele legate de tehnologii și de medii de transmisie, adică nivelurile OSI Legătură de date și Fizic.

2.1.4. Adresa IP

IP (Internet Protocol) este un protocol care asigură un serviciu de transmitere a datelor, fără conexiune permanentă. Acesta identifică fiecare interfață logică a echipamentelor conectate printr-un număr numit „adresă IP”. Versiunea de standard folosită în majoritatea cazurilor este IPv4. În IPv4, standardul curent pentru comunicarea în Internet, adresa IP este reprezentată pe 32 de biți (de ex. 192.168.0.1). Alocarea adreselor IP nu este arbitrară; ea se face de către organizații însărcinate cu distribuirea de spații de adrese. De exemplu, RIPE este responsabilă cu gestiunea spațiului de adrese atribuit Europei.

2.1.5. Port

Portul este un număr pe 16 biți care identifică în mod unic procesele ce rulează pe un sistem de calcul. Orice aplicație care realizează o comunicare în rețea va atașa un port conexiunii respective. Porturile cuprinse între 0 și 1023 sunt rezervate unor procese de sistem.

2.2. Baze de date

Aplicatia stocheaza date cu privire la useri cum ar fi : username , mesaje , id-user ,id-mesaj , conversatii , id-conversatie prin intermediul bazelor de date de tip SQL (Structured Query Language).

Baza de date SQL sau baza de date relațională este o colecție de tabele foarte structurate, în care fiecare rând reflectă o entitate de date și fiecare coloană definește un câmp de informații specific. Bazele de date relaționale sunt construite folosind limbajul de interogare structurat (SQL) pentru a crea, stoca, actualiza și prelua date. Prin urmare, SQL este limbajul de programare de bază pentru toate sistemele de gestionare a bazelor de date relaționale (RDBMS), cum ar fi MySQL, Oracle și Sybase, printre altele.

3. Arhitectura aplicatiei

3.1. Concepte folosite

- Interfata de login

Utilizatorii de vor conecta in aplicatia de client prin intermediul unui username ;

- New user

Daca utilizatorii nu au un username acestia is vor crea unul prin intermediul comenzii new user;

- Mesaje noi

Aceasta comanda permite vizualizarea mesajelor primite in timpul in care utilizatorul este deconectat;

- Mesaj nou

Prin intermediul acestei comenzi utilizatorii pot primite un mesaj catre alt utilizator ;

- Istoric mesaje

Utilizatorii pot vedea istoricul mesajelor dintr-o anumita conversatie

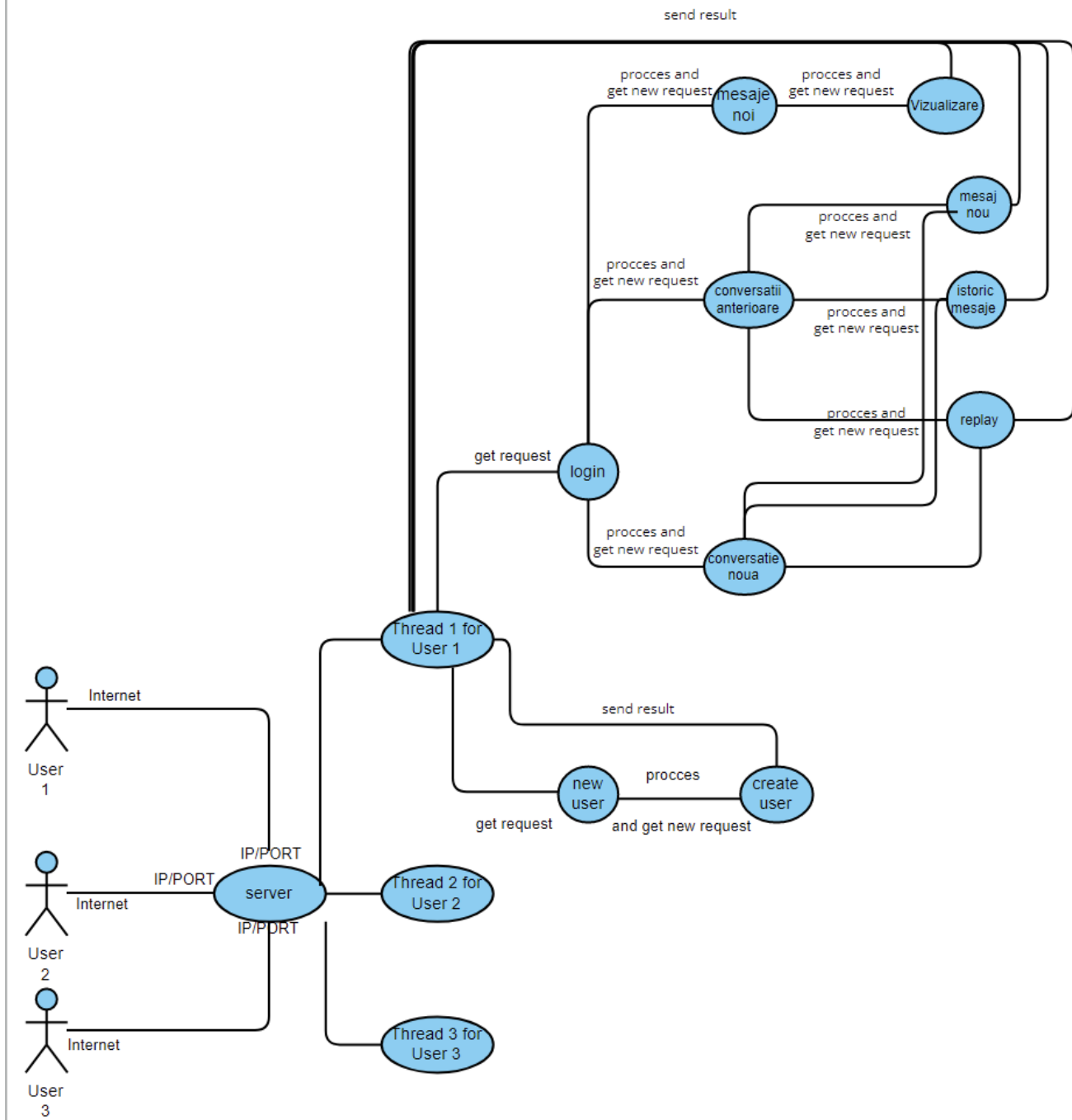
- Replay

Cu ajutorul acestei comenzi utilizatorii pot raspunde la un anumit mesaj selectat;

- Conversatie noua

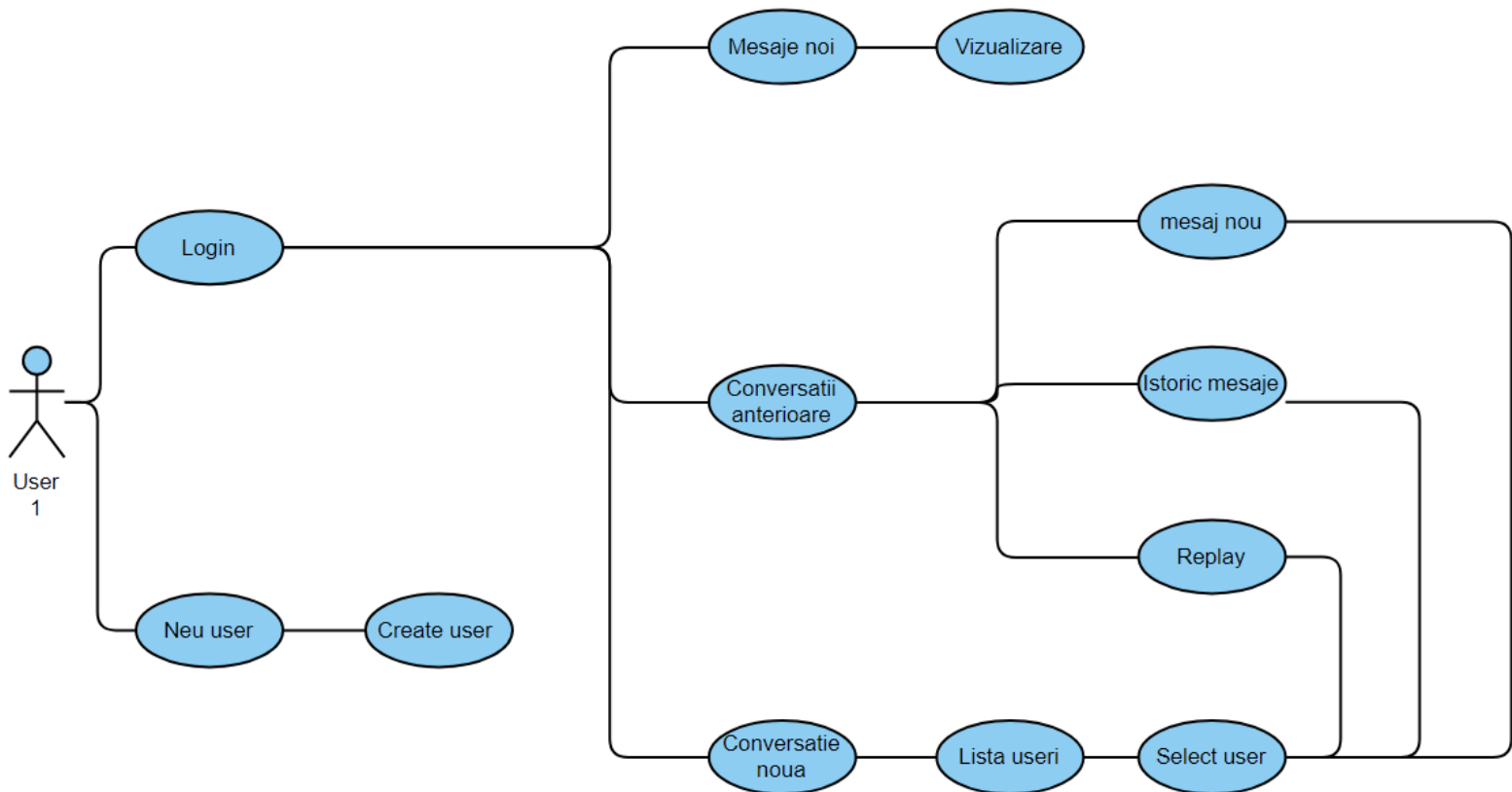
Prin intermediul acestei comenzi utilizatorii por crea o conversatie noua cu un utilizator nou;

3.2. Diagrama aplicatiei



4. Detalii de implementare

4.1. Scenarii utilizare



- Utilizatorul daca are un username valid se conecteaza prin intermediul comenzii de login;
- Daca acesta este nou in aplicatie , va apela comanda new user si isi va crea un username prin intermediul comenzii de create user;
- Dupa logare utilizator are trei optiuni:
 1. Accesarea unei conversatii anterioare ;
 2. Creerea unei conversatii noi cu un alt user , implicit selectarea acestia dintr-o lista de useri;
 3. Vizualizarea mesajelor noi primite in timpul in care utilizatorul a fost delogat;
- Dupa intrarea intr-o conversatie utilizatorul are 3 optiuni :
 1. Scrierea unui mesaj nou ;
 2. Vizualizarea istoricului unei conversatii ;
 3. Replay la un mesaj existent cu un alt mesaj;

5. Concluzii

5.1. Imbunatatiri ale aplicatiei

- Aplicatia poate fi imbunatatita prin implementarea unei interfete grafice prentu a facilita accesul, tuturor modulelor implementate , utilizatorilor neexperimentati ;
- Implementarea la nivelul modului de conectare a conectari cu o parola care sa fie case-sensitive , pentru a spori nivelul de securitate, care sa fie configurata dupa un anumit model cu o lungime prefixata;
- Codificarea mesajelor pentru ca cei care au acces la baza de data sa nu poata previzualiza mesajele utilizatorilor.
- Posibilitatea de a trimite si alte mesaje decat cele text cu ar fi : poze , video-uri , audio-uri , emoji-uri ,etc.;
- Posibilitatea d a reactiona la mesaje prin emoji-uri;
- Posibilitatea crearii unor chat-uri de tip grup unde pot lua parte la conversatie mai mult de doi utilizatori ;
- Posibilitatea de a sterge mesaje ;
- Posibilitatea de a redirectiona un mesaj ;

6. Bibliografie

1. Materiale curs
2. Materiale seminar
3. https://ro.wikipedia.org/wiki/Transmission_Control_Protocol
4. <https://learn.microsoft.com/en-us/host-integration-server/core/iterative-vs-concurrent-tcp-ip-models1>
5. https://ro.wikipedia.org/wiki/Adres%C4%83_IP
6. <https://www.ibm.com/docs/en/cics-ts/6.1?topic=protocols-tcpip-internet-addresses-ports>
7. Site pentru realizarea diagramelor: <https://online.visual-paradigm.com/drive/#diagramlist:proj=0&diagram=list>
8. <https://www.solarwinds.com/resources/it-glossary/sql-database>